Introduction to Active Server Pages (ASP)

Prerequisites: All students must have a complete knowledge of HTML

Student Objectives:

- To learn basic programming skills that apply to any modern programming language, as well as the specifics of VBScript (Visual Basic, Scripting Edition), the language in which most ASPs are written.
- To develop a complete understanding of the Active Server Pages (ASP) architecture and of the range of applications that can be developed using Microsoft Internet Information Server (IIS) and ASPs.
- To learn how to build database-driven sites, such as on-line storefronts and employee locators, using ASPs.
- To understand how ActiveX components, including Active Data Objects, integrate with ASPs, and to learn how the functionality of your ASPs can be extended using freeware, shareware, and commercial ActiveX components.

Day One

Overview of the ASP Architecture

- Requires Internet Information Server or ChiliASP + another server
- Processed server-side as an in-server process
- Must have executable permission turned on
- Can be written in VBScript, Jscript (JavaScript), or PerlScript
- Compare/contrast with CGI scripts, client-side JavaScript, and Java

Introduction to Programming, VBScript, and ASP

- What programming languages are used for and how they are typically structured
- Why is VBScript the language of choice for writing ASPs? What distinguishes it from other languages?
- An introduction to the core ideas of "object-based" programming and why object-based languages (such as VBScript) are usually much easier to program.
- Statements and scalar variables in VBScript
- How to build a simple ASP with VBScript
- **Exercise:** Building a greetings page for your hypothetical company, the WestLake Laundry Detergent Company, that automatically displays the current date and time

The Request and Response Objects

- Introduction to the Request object (which contains the data submitted by the user via a form)
- The GET and POST methods for submitting a form
- Retrieving data submitted via a form

- Introduction to the Response object (which is used to transmit a dynamically-generated page back to the browser)
- **Exercise:** Writing an ASP that takes the name a user enters in a form and displays an appropriate greeting in response

Conditionals

- Using IF-THEN-ELSE-END IF to execute one set of statements or another depending on whether a conditional is true or false. for example, you might want the ASP to generate a greetings page that says "Hello, Ms. Garcia" if the visitor is a woman or "Hello, Mr. Garcia" if the visitor is a man.
- Comparison operators (equals, does not equal, is greater than, etc.)
- **Exercise:** Enhancing the ASP from the last exercise to give a gender-specific greeting
- Using the ELSE IF condition to create multi-way branches
- **Exercise:** Enhancing the ASP from the previous exercise to ask the user to mark her/his gender if she/he did not mark one in the form

Enhancing Conditionals with AND, OR, and NOT

- The use of AND, OR, and NOT
- Using parentheses to guarantee how these conditionals are evaluated
- **Exercise:** Completing an ASP that evaluates a user's responses to an online survey

More about the Request Object and Form Processing, plus an Introduction to Cookies

- What the browser sends the server
- Environment variables
- Certificate information
- Cookies
- User-supplied information from forms
- The **ServerVariables** collection
- The **Form** and **QueryString** collections for processing form input
- The basics of cookies and their role in ASP applications
- **Exercise:** Enhancing the greetings page from the previous exercise with forms and cookies so that the page either greets the user by name or prompts her to create an account, depending on whether she has visited the page before

Overview of the Session and Application Objects

- Storing information in the **Session** object
- Overview of the **Application** object
- The role of the **global.asa** file.
- **Exercise:** Enhancing your page with a hit counter

Introduction to Using Server-Side ActiveX Components in your ASPs

- Overview of ActiveX and ASP's component architecture

- Finding and installing components
- The Ad Rotator Component
- **Exercise:** Using the Ad Rotator component to enhance your page with a random banner ad

Day Two

Sending Email from your ASPs

- Overview of the various mail components available
- How to use the free JMail component
- How "concatenation" (joining together pieces of information) is accomplished in VBScript
- **Exercise:** Creating a product order form whereby users can order products. The resulting order is emailed as text to the user and to the webmaster (which in class will be you).

Loops and Arrays in VBScript

- How arrays are used to store lists of information
- When loops are used to perform an operation for each item in an array
- While and Until loops
- **Exercise:** Enhancing the ASP from the previous exercise to send out a copy of the order to each recipient in an array of recipients

Collections in VBScript

- The differences between arrays and collections
- How <SELECT> lists are handled via VBScript in ASPs, including how to write a loop that will perform one operation on each item marked in the list.
- **Exercise:** Enhancing the ASP from the previous section so that it lists each product you select in a <SELECT> list of products

More about Loops in VBScript

- Using Next and Last to control the behavior of a loop
- For loops
- Do...while and Do...until loops

Database Review and Introduction to SQL

- Review of database concepts (tables, columns, datatypes, keys, and design issues)
- SQL Primer: SELECT, INSERT, UPDATE, DELETE
- WHERE = vs. WHERE LIKE
- Wildcards in SQL
- Using AND and OR
  Exercise: Performing database queries against a sample database by handwriting the SQL statements

Introduction to ASP-Database Connectivity via ADODB

- Overview of the ASP-database connectivity architecture, including Active Database Objects (ADODB)
- The Server object
- The **HTMLEncode()** method
- The **CreateObject()** method
- Setting up an ODBC System DSN for your database
- The **Connection** object
- Creating a **Connection** object
- Using the **Execute()** method to return and modify data
- Closing the connection
- The **RecordSet** Object
- How a **RecordSet** object is created
- How to access the fields in a record.
- **Exercise:** Completing an ASP that looks up a person in the client database by email address and returns their first and last names

Working with Multiple-Record Queries

- Writing queries that will generate more than one matching record
- Looping through and displaying the multiple records as bulleted lists or tables.
- **Exercise:** Completing a script that takes an SQL select statement entered by the user and returns the results as a table
- **Exercise:** Completing a script that allows the user to search for a given value in the field of her choice, returning the results as a table

Day Three

Subroutines and Functions in VBScript

- The roles of subroutines and functions in programming
- Demonstration of dividing a program into subroutines and functions (and why this would be done)
- **Exercise:** Making the code in the previous exercise easier to manage and reuse by dividing it into functions and subroutines

Adding Records to a Database of Orders

- Overview of techniques and precautions for adding records to a database
- **Exercise:** Modifying the order form from the exercise yesterday afternoon so that rather than sending email, it adds the order to a database of orders

Selecting and Updating Records in the Database

- Brief overview of techniques and precautions for updating records in a database, including a brief overview of transactions, lock types, and isolation levels
- **Exercise:** Building an ASP-based interface for retrieving and modifying an order

Transactions in ASP

- Why transactions are important
- Committing and rolling back transactions

- **Exercise:** Enhancing the ASP from the previous exercise to use transactions

Deleting Records from the Database

- Techniques and precautions for deleting records from the database
- **Exercise:** Completing an ASP-based interface for retrieving and canceling an order

Integrating ASPs with Microsoft Access 2000

- Benefits and limitations of Access 2000, with a discussion of when a more robust database (e.g., SQL Server, Oracle, or another true RDBMS) should be used
- Overview of how to export an ASP from Access 2000 (and the pros and cons of doing this)
- **Exercise:** Auto-generating an ASP interface to an Access 2000 product inventory database
- Time permitting, a brief overview of IDC/HTX and how to convert IDC/HTX applications to ASP